

Type-2 Gumbel 수명분포의 형상모수 변화에 따른 소프트웨어 신뢰성 비용모형의 속성에 관한 연구

양태진*

A Study on the Attributes of Software Reliability Cost Model with Shape Parameter Change of Type-2 Gumbel Life Distribution

Tae-Jin Yang*

요약 본 연구에서는 NHPP 모형을 적용한 Type-2 Gumbel 수명분포의 형상모수 변화에 따른 소프트웨어 개발비용 모형에 관한 속성을 비교, 분석하였다. 소프트웨어 고장현상을 분석하기 위하여 모수추정은 최우추정법을 적용하였고, 비선형 방정식의 계산은 이분법을 사용하였다. 그 결과, 형상모수의 변화에 따른 비용곡선의 속성을 비교하였을 때 형상모수가 클수록, 소프트웨어 개발비용이 적고 또한, 방출시간도 가장 빠르다는 것을 알 수 있었다. 본 연구를 통하여, 소프트웨어 개발자들에게 소프트웨어 형상모수 변화에 따른 개발비용을 탐색하는데 도움을 줄 수 있으며, 더불어 소프트웨어 개발비용에 관한 속성을 파악하는데 필요한 사전정보를 제공할 수 있을 것으로 기대한다.

Abstract In this study, we compare and analyze the attributes of the software development cost model according to the shape parameters change of the Type-2 Gumbel lifetime distribution using the NHPP model. In order to analyze the software failure phenomena, the parametric estimation is applied to the maximum likelihood estimation method, and the nonlinear equations are calculated using the bisection method. As a result, when the attributes of the cost curves according to the change of shape parameters are compared, it is found that the larger the number of shape parameters, the lower the software development cost and the faster the release time. Through this study, it is expected that it will be helpful for the software developers to search for the development cost according to the software shape parameters change, and also to provide the necessary information for the attributes of the software development cost.

Key Words : Laplace trend test, Life distribution, NHPP, Shape parameter, Software reliability cost model, Type-2 Gumbel distribution.

1. 서론

4차 산업혁명 시대의 급격한 도래와 함께 빅-데이터를 분석하기 위한 소프트웨어 기술도 급격히 성장하고 있다. 따라서 대용량 자료를 분석하기 위한 소프트웨어를

개발과정에서 비용관련문제는 매우 중요한 문제 중에 핵심요소가 되었다. 결국, 소프트웨어 개발과정에서 최적의 개발비용을 사전에 예측 할 수 있다면 소프트웨어 제품 비용을 경제적으로 대처 할 수 있을 것으로 판단된다. 소프트웨어 초기개발비용을 낮출 수만 있다면 소프

Funding for this paper was provided by Namseoul University year 2019

*Corresponding Author : Department of Electronic Engineering, Namseoul University(solomon645@nsu.ac.kr)

Received May 08, 2019

Revised June 11, 2019

Accepted June 11, 2019

트웨어 제품의 가격도 낮출 수 가 있음을 의미하며, 이는 소프트웨어 유지 및 보수비용에도 영향을 미칠 수 있다[1]. 지금까지 소프트웨어의 비용 발생을 사전에 탐색하기 위한 신뢰성 모형에 관한 연구가 활발히 진행되고 있고 특히, 효율적인 개발비용 모형을 구축하는 연구도 이미 오래전부터 수행되고 있다[2]. 이러한 모형의 바탕에는 비동질적 포아송 과정(Non-homogeneous Poisson process; NHPP)을 적용한 모형이 이용되고 이 모형은 결함이 발생하면 즉시 제거되며, 결함을 수리하는 과정에는 새로운 결함이 발생되지 않는다는 유한 고장의 가정을 전제로 하고 있어서 효율적인 모형으로 평가된다[3]. 이런 연구와 관련하여 Goel and Okumoto는 지수 형태의 소프트웨어 신뢰성 모형을 제안하고[4], 또한, Kim은 NHPP 신뢰성 모형에 기초하여 Type-2 Gumbel 분포의 신뢰성 속성을 분석을 하기도 하였다[5]. 본 연구에서는 소프트웨어 개발비용 모형을 기초로 소프트웨어 수명분포의 형태를 결정하는 형상모수의 변화에 따른 신뢰도 속성 분석을 수행하였다.

따라서, 본 연구에서는 소프트웨어 개발비용 모형에 유한고장 NHPP 모형을 적용한 후, Type-2 Gumbel 수명분포의 형상모수의 변화에 따른 그 속성을 비교 분석 하였다.

2. NHPP 소프트웨어 신뢰성 모형

$N(t)$ 을 시간 t 까지 탐지된 소프트웨어의 고장의 누적 횟수이고, $m(t)$ 를 이에 대한 평균값함수이고 $\lambda(t)$ 을 강도함수(Intensity function)라고 하면, 누적 고장 수 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률밀도함수(Probability density function)를 따른다고 하였다[7]. 비동질적 포아송 과정(NHPP)의 소프트웨어 신뢰성 모형은 단위 시간당 발생되어지는 고장의 횟수를 기반으로 평균 고장률 함수 $m(t)$ 를 이용해서 소프트웨어의 신뢰성을 측정하는 모형이다.

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!} \quad (1)$$

단, $n = 0, 1, 2, \dots, \infty$

(1)식에 의한 NHPP 모형의 평균값 함수 $m(t)$ 와 강도함수 $\lambda(t)$ 는 다음과 같다[11].

$$m(t) = \int_0^t \lambda(s) ds \quad (2)$$

$$\frac{d m(t)}{dt} = \lambda(t) \quad (3)$$

시간관련(Time domain) NHPP 모형들은 유한고장과 무한고장으로 분류된다. 유한고장 NHPP 모형에서는 시간(0, t]까지 발견될 수 있는 고장의 기대값을 θ 라고 하면, 유한고장 NHPP모형의 평균값 함수와 강도함수는 다음과 같다[6].

$$m(t) = \theta F(t) \quad (4)$$

$$\lambda(t) = \theta F'(t) \quad (5)$$

한편, 시간 (0, t]까지 고장이 발견되기 위한 시간 절단(Time truncated) 모형은 n 번째까지 고장시점 자료를

$$x_n = \sum_{i=1}^n t_i \quad (6)$$

(단, $i = 1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n$)

이라고 하면, 데이터 D_{x_n} 은 $\{x_1, x_2, \dots, x_n\}$ 으로 구성된다. 이와 같은 고장 절단모형에서 θ 을 모수공간이라고 표시하면, NHPP 모형의 우도함수는 다음과 같다[1, 3].

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp[-m(x_n)] \quad (7)$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$

3. 소프트웨어 개발 비용모형

일반적으로 소프트웨어 개발 비용모형은 다음과 같이 구성된다[7].

$$\begin{aligned}
 E_i &= E_1 + E_2 + E_3 + E_4 & (8) \\
 &= E_1 + C_2 \times t + C_3 \times m(t) \\
 &\quad + C_4 \times [m(t+t') - m(t)]
 \end{aligned}$$

단, E_i : 소프트웨어 개발 총비용
 E_1 은 초기 소프트웨어 개발비용을 의미하며, E_2 는 단위 시간당 일정한 소프트웨어 테스트 비용을 나타낸다.

$$E_2 = C_2 \times t \quad (9)$$

단, C_2 는 단위 시간당 비용이고, t 는 테스트 시점. E_3 는 기본 결함을 감지하고, 하나의 고장을 제거하는 비용을 의미하며 다음과 같다.

$$E_3 = C_3 \times m(t) \quad (10)$$

단, C_3 는 테스트 과정에서 발견된 하나의 고장을 제거하는 비용, $m(t)$ 는 t 시점에서 발견되어 질 수 있는 평균 고장률 함수로 고장의 기대값을 나타낸다. 또한 E_4 는 소프트웨어 시스템에 남아있는 모든 고장들을 제거하는 비용으로 다음과 같다.

$$E_4 = C_4 \times [m(t+t') - m(t)] \quad (11)$$

단, C_4 는 소프트웨어를 출시한 이후에 소프트웨어 사용 단계에서 사용자가 발견한 고장을 수정하는 비용, t' 는 소프트웨어를 출시한 이후에 소프트웨어를 고장없이 정상적으로 사용할 수 있는 유지시간을 의미한다. 여기서, C_4 는 비용 C_2 와 C_3 보다는 높다는 판단이 현실적이 될 수 있다.

또한, 소프트웨어를 방출하는 시간(t)는 소프트웨어 개발 총비용(E_i)이 최소가 되는 시점이기 때문에, 다음과 같이 유도 할 수 있다[8].

$$\frac{\partial E_i}{\partial t} = E_i' = (E_1 + E_2 + E_3 + E_4)' = 0 \quad (12)$$

4. 제안된 유한고장 NHPP Type-2 Gumbel 모형

모수추정 결과로부터 Type-2 Gumbel의 확률밀도함수(PDF)와 분포함수(CDF)는 다음과 같다.

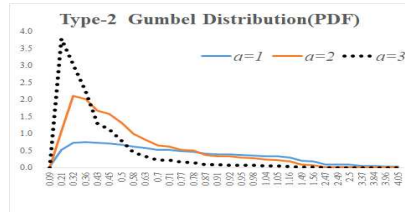


그림 1. Type-2 Gumbel 분포의 PDF
 Fig. 1. PDF of Type-2 Gumbel distribution

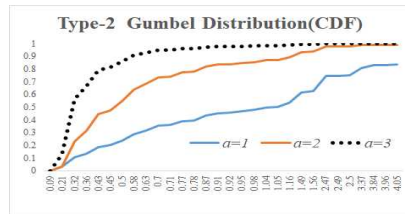


그림 2. Type-2 Gumbel 분포의 CDF
 Fig. 2. CDF of Type-2 Gumbel distribution

본 연구에서는 소프트웨어 고장수명분포의 형태를 결정하는 형상모수(a)의 변화에 따라서, 유한고장 NHPP 모형의 개발비용 속성을 비교, 분석하는 것이 목적이다. 즉, 초기 고장 형태가 가장 높다가 가장 낮은 경우($a=3$), 초기 고장 형태가 $a=3$ 인 경우보다 낮고 고장시간이 흐름에 따른 고장형태가 높은 경우($a=2$), 그리고 초기 고장 형태가 가장 낮고 고장시간이 흐름에 따른 고장형태가 가장 높은 경우($a=1$)를 구분하여 신뢰속성이 연구되었다.

또한, 유한고장 NHPP Type-2 Gumbel 모형의 강도함수와 평균값 함수는 다음과 같다[9].

$$\lambda(t|a, b) = \theta (a b t^{-a-1} e^{-bt^{-a}}) \quad (13)$$

$$m(t|a, b) = \theta e^{-bt^{-a}} \quad (14)$$

단, $a, b > 0, t \in [0, \infty]$

따라서, (7)식에 (13)식과 (14)식을 대입하여, 양변에 자연로그를 취하여 정리하면, NHPP 로그우도 함수는 다음과 같다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln \theta + n \ln a + n \ln b - (a+1) \sum_{i=1}^n \ln x_i - b \sum_{i=1}^n x_i^{-a} - \theta e^{-bx_n^{-a}} \quad (15)$$

그러므로, (15)식에서 형상모수가 각각 $a = 1, 2, 3$ 인 경우를 고려하고자 한다. 따라서, (15)식을 만족하는 모수 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 의 최우추정값을 수치 해석적 방법으로, 다음과 같이 계산할 수 있다[10].

① 형상모수 $a = 1$ 인 경우의 최우추정값

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - e^{-bx_n^{-1}} = 0 \quad (16)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial b} = \frac{n}{b} - \sum_{i=1}^n x_i^{-1} + \theta x_n^{-1} e^{-bx_n^{-1}} = 0 \quad (17)$$

② 형상모수 $a = 2$ 인 경우의 최우추정값

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - e^{-bx_n^{-2}} = 0 \quad (18)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial b} = \frac{n}{b} - \sum_{i=1}^n x_i^{-2} + \theta x_n^{-2} e^{-bx_n^{-2}} = 0 \quad (19)$$

③ 형상모수 $a = 3$ 인 경우의 최우추정값

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - e^{-bx_n^{-3}} = 0 \quad (20)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial b} = \frac{n}{b} - \sum_{i=1}^n x_i^{-3} + \theta x_n^{-3} e^{-bx_n^{-3}} = 0 \quad (21)$$

5. 소프트웨어 고장시간을 이용한 비용 속성의 분석

본 연구에서는 소프트웨어 고장시간 자료[2] 를 이용하였는데, 이 자료의 고장시간은 405 시간단위에 30 번의 고장이 발생한 자료이며, [표 1]에 요약되어 있다. 해당 자료를 가지고 유한고장 NHPP Type-2 Gumbel 신뢰성 모형의 개발비용을 비교한 후, 속성을 분석하고자 한다.

표 1. 소프트웨어 고장시간 자료[12]
Table 1. Software Failure Time Data[12]

Failure number	Failure time (hours)	Failure time (hours) × 10 ⁻²
1	9	0.09
2	21	0.21
3	32	0.32
4	36	0.36
5	43	0.43
6	45	0.45
7	50	0.50
8	58	0.58
9	63	0.63
10	70	0.70
11	71	0.71
12	77	0.77
13	78	0.78
14	87	0.87
15	91	0.91
16	92	0.92
17	95	0.95
18	98	0.98
19	104	1.04
20	105	1.05
21	116	1.16
22	149	1.49
23	156	1.56
24	247	2.47
25	249	2.49
26	250	2.50

27	337	3.37
28	384	3.84
29	396	3.96
30	405	4.05

본 연구에서는 [표 1]에 수록된 소프트웨어 고장시간 자료에 대한 신뢰도를 확보하기 위하여, 라플라스 추세검정(Laplace trend test)을 사용하였다. 일반적으로 라플라스 요인 추정 값이 -2와 2사이에 분포하면 극단적인 값이 존재하지 않고 안정적이기 때문에 자료에 대한 신뢰성을 가진다고 하였다. 본 연구에서의 라플라스 추세검정의 결과는 [그림 3]에서 보여주듯이 -2와 0사이에서 분포하여 [표 1] 자료를 사용하는 것이 유효하다고 판단된다[11].

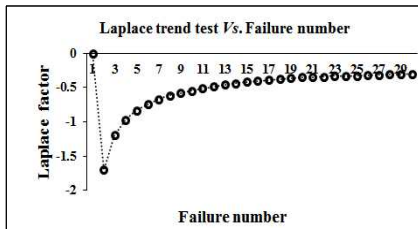


그림 3. 라플라스 검정의 추정결과
Fig. 3. Estimation results of Laplace test

또한, 모수추정은 최우추정법 (Maximum Likelihood Estimation ; MLE)을 이용하였고, 모수 추정의 수렴성을 위하여 고장시간 자료를 수치변환 ($Failure\ time \times 10^{-2}$)하여 적용하였다[12]. 비선형 방정식의 풀이는 이분법(Bisection method)을 사용하였고, 그 결과는 [표 2]와 같다.

표 2. 형상모수 변화에 따른 모수 추정값
Table 2. Parameter estimation value according to the shape parameter change

Shape parameter	MLE	
	$a = 1$	$\hat{\theta} = 35.876$
$a = 2$	$\hat{\theta} = 30.277$	$\hat{b} = 1.507 \times 10^{-1}$
$a = 3$	$\hat{\theta} = 30.001$	$\hat{b} = 1.885 \times 10^{-2}$

본 연구에서는 실제 개발환경과 유사한 조건을

주고, 소프트웨어를 출시한 이후에 소프트웨어 사용자가 발견한 고장비용 측면(C_1)에서 변화를 준 상황으로 [가정 1]과 [가정 2]와 같은 조건으로 소프트웨어 개발비용의 속성을 분석하고자 한다.

[가정1]

$$E_1 = 5\$, c_2 = 5\$, c_3 = 1.5\$, c_4 = 10\$, t' = 10 \quad (22)$$

[가정 1]과 같은 조건을 적용한 비용 곡선의 시뮬레이션 결과는 [그림 4]와 같다. 이 그림에서 비용 패턴의 추세는 초기단계는 급격하게 감소한 후, 일정한 추세를 유지하다가 고장시간이 지나는 후반부에 가서는 점진적으로 증가하는 추세를 보이고 있다. 그 이유는 초기단계에서 소프트웨어 시스템에 남아있는 결함의 수는 결함을 제거하는 과정에서 급격히 줄어들지만, 초기단계를 지난 후 남아 있는 결함이 발견될 확률은 점점 줄어들기 때문이다. 따라서, 후반 단계에서 소프트웨어에 남아있는 결함이 탐색될 확률은 감소하기 때문에, 상대적으로 결함이 발견될 시간이 길고, 고장을 제거하는 비용도 함께 증가하게 된다. 그러므로, 이러한 비용곡선의 추세를 이용하여 비용이 최소가 되는 시점을 사전에 찾으면, 최적의 소프트웨어 방출(출시) 시간도 함께 예측 할 수 있다.

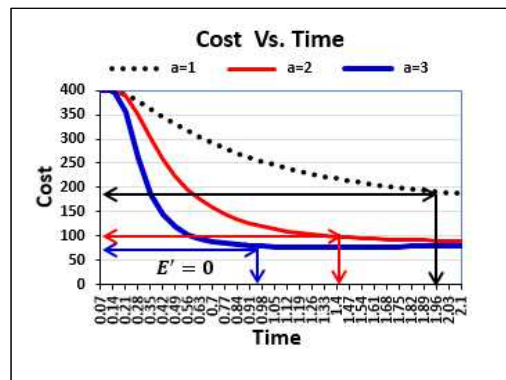


그림 4. [가정 1] 조건을 적용한 개발비용 곡선
Fig. 4. The development cost curve applied to the condition of [Supposition 1]

[그림 4]의 시뮬레이션 결과를 세부적으로 분석해보면, 형상모수 $a=1,2,3$ 일 때, 모든 초기 추세는 비

용이 급격히 감소한 후, 일정 추세를 보이다가 고장시간이 지나는 후반부에서 점차 증가하는 패턴을 보이고 있다. 즉, 형상모수 $a=3$ 일 때 개발비용이 가장 적고, 방출시간도 빨라서 가장 우수함을 알 수 있었다. 따라서, 이 경우에는 소프트웨어 방출 이전에 결함을 감소시킬 수 있도록 실제 운영단계보다는 사전 테스트 단계에서 가능한 모든 결함들을 제거해야 한다.

[가정2]

$E_1 = 5\$, c_2 = 5\$, c_3 = 1.5\$, c_4 = 20\$, t' = 10$
 (23) [가정 1]과 비교하여, 다른 조건은 모두 동일하지만, 실제개발환경과 유사한 조건을 부여하기 위하여, 소프트웨어 출시 이후에 실제 소프트웨어 운영 단계에서 사용자가 발견한 결함수정 비용인 C_4 의 변화를 준 경우이다. 해당 시뮬레이션 결과는 [그림 5]와 같다.

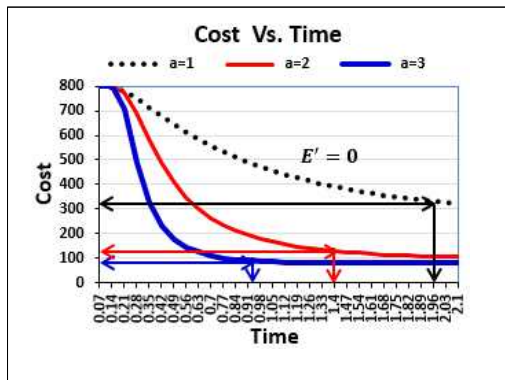


그림 5. [가정 2] 조건을 적용한 개발비용 곡선
 Fig. 5. The development cost curve applied to the condition of [Supposition 2]

[그림 5]의 시뮬레이션 결과를 분석하면, 실제 소프트웨어 운영 단계에서 사용자가 발견한 결함수정 비용 C_4 를 10\$에서 20\$로 2배 상승시키면 개발비용은 증가하지만, 반대로 소프트웨어 방출시간에는 변함이 없다는 것을 알 수 있었다. 즉, 실제 개발환경과 같은 비용 조건에 변화를 주어도, 형상모수 a 가 클수록 개발비용이 적고, 방출시간도 빨라서 가장 우수함을 알 수 있었다. 그러므로, 소프트웨어 방출

이전에 결함을 감소시킬 수 있도록 실제 운영단계보다는 사전 테스트 단계에서 가능한 모든 결함들을 제거해야 한다.

6. 결론

실제 소프트웨어 운용 과정이나 소프트웨어 테스트 작업에서 발생할 수 있는 고장시간에 대한 신뢰성을 소프트웨어 개발비용에 적용한 후 모형화하면, 비용의 속성을 더욱 효율적으로 파악할 수 있다. 그러므로, 본 연구에서는 실제 소프트웨어 고장시간 자료를 바탕으로, 유한고장 NHPP 신뢰성 모형에서 Type-2 Gumbel 수명분포의 형상모수 변화에 따른 비용 모형의 속성을 비교, 분석을 하였다.

본 연구의 결과를 다음과 같이 요약할 수 있다.

첫째, 기본 조건하(가정1)에서 비용곡선은 초기단계에는 현저히 감소한 후, 일정한 패턴을 보이다가 고장시간이 지나는 후반단계에는 점진적으로 증가하는 패턴을 보였다. 그 이유는 남아있는 결함의 수는 결함을 제거하는 과정에서 줄어들기 때문에 잔존 결함이 발견될 확률은 점점 낮아져서, 결국 비용이 상승하기 때문임을 알 수 있었다.

둘째, 실제 개발환경과 동일한 조건을 주기 위해서 소프트웨어 출시이후에 소프트웨어 운용단계에서 사용자가 발견하는 결함수정 비용을 증가시키면, 개발비용은 같이 증가하였지만, 반대로 방출시간에는 변함이 없다는 것을 알 수 있었다.

셋째, 본 연구에서 적용한 조건을 종합적으로 분석한 결과, 고장분포의 형태를 결정하는 형상모수 a 가 클수록 우수한 모형임을 알 수 있었다.

결론적으로, 본 연구결과를 이용하면 소프트웨어 개발자 및 사용자들에게 최적의 소프트웨어 방출 시간과 가장 경제적인 개발비용을 예측하는데 필요한 사전 정보를 제공할 수 있다고 판단된다. 더불어 추후에 같은 형태의 고장 분포를 갖는 다른 신뢰성 모형들과의 소프트웨어 개발비용과 방출시간의 속성 분석을 통해 최적의 신뢰성 비용 모형을 찾아내는 후속 연구가 필요하겠다.

REFERENCES

- [1] K. Y. Song, I. H. Chang, H. Pham, "A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments", Applied Science, Vol.7, No.983, pp.1-16, 2017.
- [2] T. J. Yang, "A characteristic study on the software development cost model based on the lifetime distribution following the shape parameter of Type-2 Gumbel and Erlang distribution", The Journal of Korea Institute of Information Electronics and Communication Technology, Vol 11, No 4, 2018, pp.460-466.
- [3] Gokhale, S. S. and Trivedi, K. S. A, "time/structure based software reliability model", Annals of Software Engineering. 8, pp. 85-121. 1999.
- [4] Goel A L, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", IEEE Transactions on Software Engineering, Vol.28, pp.206-211, 1978.
- [5] H. C. Kim, "A Comparative Study on Software Reliability Models with Shape Parameter of Type-2 Gumbel Life Distribution", International Journal of Soft Computing" Vol.12, No.5-6, pp.351-354, 2017.
- [6] T. J. Yang, "A Comparative Software Development Cost Model Considering the Change in the Shape Parameter of the Erlang Distribution", The Journal of Korea Institute of Information, Electronics, and Communication Technology" Vol.9, No.6, pp.566-572, 2016.
- [7] Ye Zhang, and Kaigui Wu, "Software Cost Model Considering Reliability and Time of Software in Use", Journal of Convergence Information Technology, Vol.7, No.13, pp.135-142, 2012.
- [8] T. J. Yang, "A Software Reliability Cost Mode Based on the Shape Parameter of Lomax Distribution", The Journal of Korea Institute of Information Electronics and Communication Technology, Vol 9, No 2, 2016, pp. 171-177.
- [9] H. C. KIM, "The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Rayleigh and Burr Type", Journal of Korea society of digital industry and information management, Vol.10, No.2, pp.1-11, 2014.
- [10] R. Satya Prasad, K. R. H. Rao and R.R.L Kantha, "Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC", International Journal of Computer Applications (0975-8887) Vol.21, No.7, pp.1-5, 2011.
- [11] T. J. Yang, "The Performance Analysis Comparative Study depend on Software Reliability Model and Curve Regression Model", Medwell Journals, Vol.12, No.5, pp.313-317, 2017.
- [12] Kuo, L. and T. Yang, "Bayesian computation of Software Reliability." Journal of Comput. Graphical Stat.4: 65-82, 1995.

저자약력

양 태 진 (Tae-Jin Yang)

[정회원]



- 1992년 2월 : 한양대학교 전자공학과 공학석사
- 1995년 2월 : 한양대학교 전자공학과 공학박사(수료)
- 1986년 3월 ~ 1992년 2월 : 현대 그룹 신규사업부 기술과장
- 1993년 3월 ~ 2013년 12월 : 호서대학교 호서전문학교 정보통신과 교수
- 2014년 3월 ~ 현재 : 남서울대학교 전자공학과 교수

<관심분야> 소프트웨어 신뢰성공학, 인공지능 (Artificial-Intelligent), Intelligent-Network & Network-Security